EURO J Transp Logist (2019) 8:119–145
https://doi.org/10.1007/s13676-017-0114-7

RESEARCH PAPER

Integrating long-haul and local transportation planning: the Service Network Design and Routing Problem

Juliette Medina[1,3] · Mike Hewitt[2] · Fabien Lehuédé[1] · Olivier Péton[1]

Received: 15 November 2016 / Accepted: 26 December 2017 / Published online: 18 January 2018

Abstract We introduce a new optimization problem, the Service Network Design and Routing Problem that integrates long-haul and local transportation planning decisions. Such a problem is particularly important for consolidation carriers that face customer demands for fast delivery and thus must synchronize the different levels of their transportation operations. We present two formulations of the problem: (1) a route-based formulation that allows for the modeling of a rich set of rules governing local delivery routes at the expense of increased instance size and computational solve time, and, (2) an arc-based formulation that can be solved more quickly but has less modeling power. We solve each with a Dynamic Discretization Discovery algorithm that was recently proposed and designed for solving Service Network Design problems that require the precise modeling of time. With an extensive computational study, we examine the benefits of each formulation.

✉ Fabien Lehuédé
fabien.lehuede@imt-atlantique.fr

Juliette Medina
j.medina@4snetwork.com

Mike Hewitt
mhewitt3@luc.edu

Olivier Péton
olivier.peton@imt-atlantique.fr

[1] IMT Atlantique, LS2N, UMR CNRS 6004, Nantes, France

[2] Quinlan School of Business, Loyola University Chicago, Chicago, USA

[3] CRC Services, Rueil Malmaison, France

Keywords Service Network Design · Vehicle Routing Problem

1 Introduction

In many cases, a transportation company is tasked with transporting a shipment that requires most, if not all of a delivery vehicle's capacity. In such cases, the shipment is large enough to generate revenues, such that it is economical to route the vehicle directly from the origin of the shipment to its destination. Ground carriers that focus their operations on these kinds of shipments are often referred to as truckload carriers.

However, there are other cases, wherein shipments are small relative to the capacity of the delivery vehicle. In these cases, the carrier is faced with a different routing problem, as the revenue generated by a single shipment is too small to cover the costs of delivering that shipment directly from origin to destination. At the same time, the size of the shipment relative to trailer capacity presents the opportunity to load multiple shipments, each with a potentially different origin and final destination, into one vehicle. As a result, carriers route these shipments through a network of terminals to consolidate multiple shipments into one vehicle. Carriers that focus on these shipments are often referred to as consolidation carriers, and typically participate in one of the two industries: (1) Small package (e.g., UPS, FedEx, DHL) and (2) Less-than-truckload (LTL) freight (e.g., XPO Logistics, UPS, Saia).

This network of carrier terminals, called *consolidation network*, facilitates the consolidation of shipments for longer, inter-city or inter-regional, movements. However, shipments must also be transported from the customer origin to a terminal in this network associated with that city or region (and similarly from a terminal in this network to the customer destination). Thus, consolidation carriers are faced with the challenge of routing shipments and resources (e.g., vehicles) on two levels and operations researchers have primarily studied methods that optimize the operations on a single level. In the first level, carriers determine routes that originate at a terminal in the consolidation network. These routes enable vehicles to pick up shipments at customer origins for transportation through the consolidation network as well as deliver shipments that have been transported through the consolidation network to customer destinations. This first level is often referred to as the *local distribution* (or *local delivery routes*) and is typically modeled with a variant of the classical Vehicle Routing Problem with Time Windows (Toth and Vigo 2014) or of the Pickup and Delivery Problem (Berbeglia et al. 2007; Parragh et al. 2008). Then, in the second level, carriers must route shipments (and resources such as vehicles) through the consolidation network to a terminal from which the local delivery routes can then transport the shipment to the customer's destination. This second level is often referred to as the *long-haul* operation, and is typically modeled with a variant of the Service Network Design Problem (SNDP) (Crainic 2000).

As such, each customer shipment often travels on three vehicle routes from origin to destination. The first vehicle route transports the shipment from the customer origin to an origin terminal in the consolidation network. The second vehicle route transports the shipment from that origin terminal to a destination terminal in the consolidation network. Finally, the third vehicle route transports that shipment from that destination terminal in the consolidation network to the customer destination. As these three routes all carry the same shipment, they in turn require synchronization. For example, it must be that the second route does not depart from the origin terminal before the shipment arrives there. Similarly, the third route must not depart from the destination terminal before the shipment arrives there.

Historically, to avoid modeling this synchronization explicitly and thus decompose the problem into a separate VRP and SNDP, operation researchers have made assumptions regarding the timing of local distribution. For example, it is common to assume that the route for a shipment through the consolidation network must arrive at the destination terminal by a certain time in the morning (to model that it must be available for local routes before they depart). Similarly, it is common to assume that a shipment will not be available at the origin terminal for routing through the consolidation network until after a certain time in the evening (to model that it will not be available for routing in the consolidation network until after the local routes have returned).

However, both LTL and small package carriers are seeing increasing customer demands for short transit times (sometimes called service standards). LTL carriers are seeing these demands in part due to their role in supply chains and the rise of just-in-time manufacturing approaches. Small package carriers are seeing these demands in part due to their role in eCommerce. This paper is motivated by an industrial collaboration with a logistics provider in France that promotes the consolidation of the goods of retail suppliers from manufacturing facilities to retail stores in city areas, through a network of logistics hubs called Collaborative Routing Centers (CRCs). This logistics provider sees the synchronization of routes at CRCs as a key element to reducing costs and satisfying customer delivery deadlines. In general, for carriers that are considering offering the same-day delivery service of shipments between cities or regions, the assumptions mentioned above, regarding the timing of events, limit the routing options considered by the resulting optimization methods. Such limitations can be costly and/or lead a carrier to forego offering the services its customers desire.

As such, in this paper, we present two mathematical models and solution methods for integrated long-haul and local transportation planning. We propose these models and methods to assist carriers with the design of transportation systems that are able to meet customer demands for short service standards in a cost-effective manner. These models integrate the two levels of routing decisions, and explicitly recognize the need for synchronization of routes. These models are derived from the time-expanded network-based service network design model presented in Boland et al. (2017). To effectively accommodate short transit times, we show how these new models can also be solved by the Dynamic Discretization Discovery (DDD) solution approach presented in that paper. As that method is an exact solution approach, the solution approaches we present are exact as well.

While the two proposed models extend the model presented in Boland et al. (2017), they differ in the degree to which they can capture operational realities. The first extension models the vehicle routing problem corresponding to the local distribution with route-based variables. Thus, it can capture all operational realities that can be encoded in a variable representing a route. For this extension, we treat DDD as a black box, and limit our scope to instances of the extended model, wherein routes are enumerated a priori (as opposed to generated on the fly via a column generation-based method). To avoid the computational difficulties associated with enumerating routes, the second extension models the vehicle routing problem associated with the local distribution with arc-based variables. While this model does not suffer from the explosion in number of variables seen in the route-based model, it implies modifications to the model for which DDD was initially proposed to solve. Furthermore, the operational realities it can capture are limited to those that can be expressed by the proposed arc-based VRP formulation.

Fundamentally, we believe that this paper presents multiple contributions. First, it presents multiple exact solution approaches for a problem that is of critical importance to both consolidation carriers in general, as customer is expressing greater demands for short transit times, and a specific carrier in France. These solution approaches are the first to capture the timing of activities at the precision that is necessary when synchronizing transportation moves in the presence of short transit times. Second, at the heart of these solutions, approaches are novel formulations of the problem that illustrate how it can be modeled as a variant of the SNDP. Third, by formulating the problem as a variant of the SNDP, we illustrate how the DDD solution approach, which was originally presented in the context of solving an SNDP with few complicating constraints, can be extended to variants that model operational realities more accurately. Finally, with an extensive computational study, we assess the effectiveness of each solution approach, their sensitivity to instance characteristics, and validate the solution of the new models proposed in this paper.

This paper is organized as follows. Section 2 presents the relevant literature, contrasting it with the work presented in this paper. Section 3 then describes the problem that we wish to study in greater detail. Section 4 presents the two optimization models that we have developed for this problem, whereas Sect. 5 details how we solve them. Then, in Sect. 6, we present the results of an extensive computational study, and in Sect. 7, we conclude and present avenues for future research.

2 Literature review

While the SNDP seeks to route shipments from an origin terminal to a destination terminal through a network of consolidation terminals, the VRP with Time Windows seeks to deliver shipments from a terminal (often called a *depot*) to customer locations within their respective time windows. Both of these problems are well studied (see Toth and Vigo 2014; Crainic 2000) and efficient solution methods, both exact and heuristic, have been developed for each. However, to our knowledge, SNDP and the VRPTW are typically modeled and solved independently; no previous work has been published on the integration of these two optimization problems. In this section, we review the most relevant works that integrate the SNDP and routing problem. We focus on the main features of the joint problem, namely the consolidation of shipments, the transfers of goods from one vehicle to another, and the time constraints related to the synchronization of routes.

In the SNDP, vehicle routing aspects have been recently considered on the long-haul network to integrate the management of assets (including trucks and trailers) in optimization models. In the tactical design of load plans, this is mainly integrated as balancing constraints on the number of vehicles entering and leaving each terminal (see, for instance, Pedersen et al. 2009; Jarrah et al. 2009; Erera et al. 2013a). More recently, vehicle management has been integrated through the design of cycles in scheduled SNDP by Andersen et al. (2009a, b) and Crainic et al. (2016). The goal is to design a cyclic planning which is motivated by the integration of various modes of transportation, including regular lines (such as railroads). Closer to an operational setting, Erera et al. (2013b) focus on determining the routes and schedules of drivers, trailers, and tractors to perform a given load plan. In these contributions, the proposed models and algorithms rely on a discretization of time which allows the authors to model the problems on time–space networks. In this discretization, time period of a few hours are acceptable in a tactical context but absolutely not for solving real-life vehicle routing problems. One of the most detailed examples of discretization is given by Erera et al. (2013b), who suppose an hourly time granularity for a time horizon of one week, but the method supposes that the load plan is given before determining the routes. The latest work of Boland et al. (2017) propose a dynamic construction of the time–space network which allows the solution of the SNDP with a 1 min granularity. This algorithm has not yet been extended to integrate assets management.

In vehicle routing, transfers between vehicles have been studied in the vehicle routing problem with cross-docking (Wen et al. 2008), the pickup and delivery problem with cross-docking (Petersen and Ropke 2011), the pickup and delivery problem with transfers (Mitrović-Minić and Laporte 2006; Cortés et al. 2010), or the two-echelon VRP (Crainic et al. 2009). In these problems, the temporal synchronization of routes is a challenge both for exact and heuristic methods (Drexl 2012). The VRP with cross-docking considers one compulsory cross-dock where goods from pickup routes can be transferred between vehicles to design delivery routes. This problem has been solved with several meta-heuristics (Wen et al. 2008; Tarantilis 2013; Morais et al. 2014; Grangier et al. 2016b) or branch-and-price (Santos et al. 2011). The pickup and delivery problem with cross-docking extends the VRP with cross-docking by considering that transiting through the cross-dock is optional. It has been solved heuristically by Petersen and Ropke (2011) and with branch-and-price by Santos et al. (2013). Routing through several cross-docks or transfer points is considered in Dondo and Mendez (2014). In this paper, a commodity can be routed from its origin to its destination through a long-haul trip between two cross-docks where pickup routes and delivery routes are performed. However, to handle synchronization in the proposed branch-and-price approach, all pickup routes are supposed to end before all transfers between cross-docks, which in turn are supposed to end before the beginning of all delivery routes. The pickup and delivery problem with transfers also integrates transfer possibilities at several transfer points in the design of pickup and delivery routes. Mitrović-Minić and Laporte (2006), Cortés et al. (2010), Qu and Bard (2012), and Masson et al. (2013a). An efficient handling of synchronization in heuristics has been proposed by

Masson et al. (2013b). Nevertheless, the pickup and delivery problem with transfers does not integrate long-haul routes and systematic consolidation aspects. Finally the two-echelon VRP has been studied in city logistics. It considers that all commodities originate at a single location and that they should be transferred from large vehicles to small vehicles at satellites locations before being delivered to customers. Up to now, only two meta-heuristics (Grangier et al. 2016a; Anderluh et al. 2017) and one exact method (Dellaert et al. 2016) integrate time windows on customers and temporal synchronization between vehicles. For a larger survey on routing with intermediate facilities, we refer to Guastaroba et al. (2016).

In conclusion, the integrated planning of long-haul and local transportation is an original problem which has not received much attention from the scientific community. To our knowledge, no previous work has been performed on this subject with the proposed flexibility on time.

3 Problem description

We introduce the Service Network Design and Routing Problem (SNDRP), which seeks to transport a set of shipments from a set of supplier locations (e.g., a warehouse or manufacturing facility) to customer locations (retail stores) through a consolidation network. We illustrate the problem that we solve in Fig. 1.

We assume an operating network in which facility location and allocation decisions have already been made. Specifically, each customer is assigned to a single terminal from which its shipments are delivered and each supplier is assigned to a single terminal to which it sends goods. We also presume that these terminals only have outgoing flow of goods. We assume direct shipments from suppliers to terminals, but allow for delivery routes that visit multiple customers.

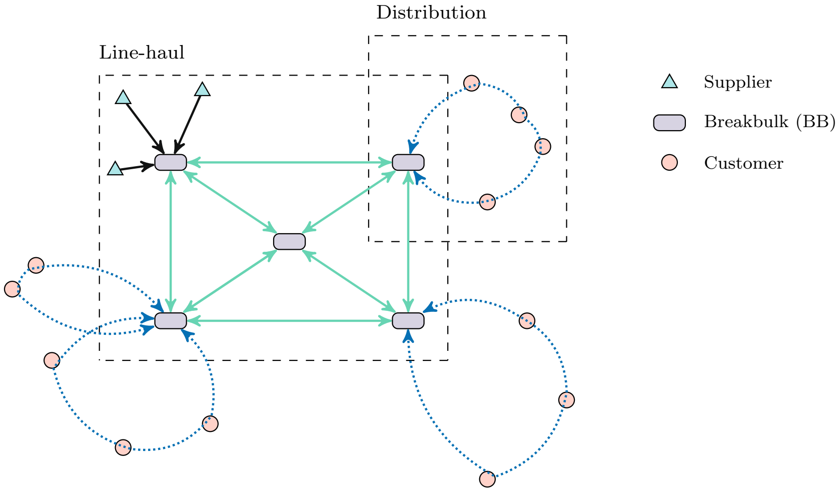


Fig. 1 Representation of the transportation network

In this paper, we call *line-haul network* the network composed of consolidation network and the supplier locations. These assumptions directly arise from our case study, where many customers are delivered from a limited number of possibly distant supplier locations.

Each shipment is defined by a supplier location (origin), a customer location (destination), a quantity, an earliest time at which it can be picked up at the supplier location, and a latest time by which it must be delivered to the customer location. The shipments are transported in the line-haul network, through which they are routed to the terminal associated with their customer location. From there, local carriers pick up the shipments and deliver them to their ultimate destination. We refer to terminals in the line-haul network as breakbulks (BB). For a given shipment, we refer to terminals that are not associated with the customer location as intermediate terminals, and the terminal that is associated with the customer location the destination terminal.

When freight arrives at an intermediate terminal, it is unloaded from inbound transportation, consolidated with other freight, and loaded onto outbound transportation to another terminal in the line-haul network. Transportation between terminals in a line-haul network may occur for many reasons, including (re)-positioning transportation resources such as trailers for future moves and drivers returning to their home terminal. However, in this paper, we only consider transportation that occurs to move freight. To transport freight between two terminals in the line-haul network, we must ensure that there is sufficient capacity. We presume a unit of capacity (e.g., the volume that a single vehicle can transport) and choose how many such units we will pay for, with each unit coming at a cost. In this sense, transportation costs are fixed, as we pay for a certain amount of capacity, regardless of how much of it is used. We presume that these costs (as well as the times required for transportation) are proportional to the distance between terminals. Upon arrival at its destination terminal, a shipment waits until a local carrier can pick it up. In a sense, this layer of the problem can be modeled as a Service Network Design Problem, with a commodity representing a shipment that has a given supplier location as its origin and the terminal associated with the customer location as its destination. However, while each shipment has a given available time, there is no fixed due time at the destination terminal, as this is determined by the timing of the local distribution.

Each destination terminal serves as a depot for a fleet of delivery vehicles. We assign each vehicle a set of shipments to deliver, and then derive a route that begins at the depot, visits the destination of each shipment at a time before it is due, and then returns to the depot. As each shipment has a size, we must ensure that the assignment does not violate the vehicle's capacity. Each move along this route comes at a cost, again independent of the volume being transported, and requires a length of time. We also present one route-based formulation of the problem that considers rules that dictate the structure of the route, e.g., labor and safety regulations, limit on the total duration of a route.

In a sense, this layer of the problem can be modeled as a Vehicle Routing Problem with Time Windows. However, the time at which the route may begin is not known, as it depends on the times at which shipments are delivered to the terminal. As such, for each shipment, we must synchronize the timing of its delivery to its destination terminal with the departure of the local delivery route that delivers it. In addition, we must do so in a way that minimizes the total transportation costs (both costs within the line-haul network and those incurred at the local delivery route) while ensuring that each shipment is picked up at its supplier after it is available and delivered to its customer before it is due. Finally, we presume that all problem parameters (transportation costs and times, vehicle capacities, and shipment sizes and available/due times) are known with certainty.

To summarize, the SNDRP consists of determining the number of vehicles between suppliers and terminals and between terminals on the line-haul network, the load plans of the vehicles, the delivery routes from each destination terminal, such that: each shipment is routed from its origin to its destination in its time window, and the sum of all transport costs are minimized. In the next section, we present two optimization models that we have formulated of this problem.

4 Optimization models

A key ingredient of the SNDRP is time. Each shipment has an earliest time at which it can be picked up at a supplier and a latest time at which it can be delivered to a customer. In addition, each transportation move requires time. However, most importantly, we need to synchronize the delivery of a shipment to its destination terminal in the line-haul network with the departure of the local delivery route. As such, to model time we use a time–space network, a graph wherein nodes encode both a physical location and a time window during which activities occur at that location. By routing freight on a time–space network, we ensure the synchronization that we need, both in time and space.

In this section, we present two formulations for the SNDRP, both of which formulate the problem as a type of SNDP on a time–space network. The first is similar to the extended, route-based, formulations one sees in the literature on the VRP (Desaulniers et al. 2006), wherein a variable encodes the choice of an entire route. This formulation also allows us to model a rich set of rules governing local delivery routes. The second is similar to the compact, arc-based, formulations one sees in the literature on the VRP (Toth and Vigo 2014).

We first introduce notations and the SNDP model that are used for both formulations. The two SNDRP formulations are presented in Sects. 4.3 and 4.4.

4.1 Notation

We first discuss how we model the physical network through which shipments are routed. We let $D = (N, A)$ represent a physical network with node set N and arc set A. The node set is comprised of three sets of nodes: N_{c}, which denotes the set of customer locations, N_{bb}, which denotes the set of breakbulk terminals, and N_{s}, which denotes the set of supplier locations. As each customer location has an associated breakbulk, we let $N_{\mathrm{c}}(i), i \in N_{\mathrm{bb}}$ denote the set of customers associated with breakbulk i. Conversely, we let $\mathrm{bb}(j)$ denote the one breakbulk associated with customer $j \in N_{\mathrm{c}}$.

The arc set A contains arcs that model moves between different types of nodes. Specifically, the arc set contains arcs between a supplier node and a breakbulk, e.g., of the form $(i,j), i \in N_{\mathrm{s}}, j \in N_{\mathrm{bb}}$. The arc set also contains arcs that model movements between two breakbulks and are of the form $(i,j), i,j \in N_{\mathrm{bb}}$.

Associated with each arc $a = (i,j) \in A$ is a travel time tt_{ij}, a fixed cost f_{ij}, a variable cost c_{ij}, and a capacity u_{ij}. As the time spent loading and unloading freight can have a critical impact on whether it is delivered on time, we embed these times in the travel time of each arc. In short, the network $D = (N, A)$ encompasses all the locations depicted in Fig. 1, and in both formulations serves to model activities in the line-haul network and in the local delivery routes.

From the physical network, we define a time–space network $D_{\mathcal{T}} = (N_{\mathcal{T}}, H_{\mathcal{T}} \cup A_{\mathcal{T}})$, where physical nodes and arcs are associated with time(s). We let $T_i = \{t_1^i, \ldots, t_{n_i}^i\}$ represent the set of all time points at node $i \in N$ and $T = \cup_{i \in N} T_i$. Then, the arc set A_t contains arcs of the form $((i,t),(j,\bar{t}))$ where arcs (i,j) belongs to the set of physical arcs A, $t \in T_i$ and $\bar{t} \in T_j$. In addition to arcs derived from those in A, we model the opportunity for shipments to wait at a terminal, either for later transportation to another terminal in the line-haul network or pickup by a local delivery route. To do so, we let the arc set $H_{\mathcal{T}}$ represent the set of *waiting arcs* of the form $((i, t_k^i), (i, t_{k+1}^i))$ $\forall i \in N$ and $k = 1, \ldots, n_{i-1}$.

The set of customer shipments is denoted by K. Each shipment must be picked up at its origin node $o_k \in N_{\mathrm{s}}$ no earlier than e_k, and delivered to its destination node $d_k \in N_{\mathrm{c}}$ no later than l_k. We also presume that associated with each shipment is a size q_k. In both formulations, a shipment will be modeled as a commodity whose origin node in the time–space network is node (o_k, e_k) and whose destination node in the time–space network is node (d_k, l_k).

4.2 Service network design model

As both of our formulations are based on the SNDP, we next present that model, using the notation just presented. We recall that the SNDP seeks to route a given set of commodities, each from its origin to its destination, through a directed network. To support such routes, one must install (potentially multiple) copies of each arc, which corresponds to choosing a level of capacity. Each installation yields capacity but also comes at a cost. There are many variants of the SNDP, including whether a commodity must follow a single path from its origin to its destination or may instead be “split” across multiple paths. In this work, we focus on the variant wherein a commodity must follow a single path.

In the SNDP, commodities travel between terminals on the line-haul network. Regarding our problem, suppliers play the role of terminals and are the origins of commodities. Destinations are breakbulks. As we focus on a problem wherein a commodity must follow a single path from its origin to its destination, we let the binary variable $x_{ij}^{kt\bar{t}}$ represent whether commodity k uses arc $((i,t),(j,\bar{t})) \in A_{\mathcal{T}}$. In the SNDP, a variable cost c_{ij} is paid for each unit of flow on arc $(i,j) \in A$. Regarding capacity, we let the variable $y_{ij}^{t\bar{t}}$ represent the number of vehicles that travel on arc $((i,t),(j,\bar{t})) \in A_{\mathcal{T}}$. The fixed cost, f_{ij}, and capacity level, u_{ij}, are associated with these variables. In short, the SNDP seeks to

$$\min Z = \sum_{((i,t),(j,\bar{t}))\in A_{\mathcal{T}}} f_{ij}\, y_{ij}^{t\bar{t}} + \sum_{k\in K} \sum_{((i,t),(j,\bar{t}))\in A_{\mathcal{T}}} c_{ij}\, q_k\, x_{ij}^{kt\bar{t}} \tag{1}$$

s.t.

$$\sum_{((i,t),(j,\bar{t}))\in A_{\mathcal{T}}\cup H_{\mathcal{T}}} x_{ij}^{kt\bar{t}} - \sum_{((j,\bar{t}),(i,t))\in A_{\mathcal{T}}\cup H_{\mathcal{T}}} x_{ji}^{k\bar{t}t} = \begin{cases} 1 & \text{if}(\mathrm{i},\mathrm{t}) = (\mathrm{o_k},\mathrm{e_k}) \\ -1 & \text{if}(\mathrm{i},\mathrm{t}) = (\mathrm{d_k},\mathrm{l_k}) \quad \forall k \in K, \forall (i,t) \in N_{\mathcal{T}} \\ 0 & \quad \text{otherwise.} \end{cases} \tag{2}$$

$$\sum_{k\in K} q_k x_{ij}^{kt\bar{t}} \le u_{ij} y_{ij}^{t\bar{t}} \quad \forall ((i,t),(j,\bar{t})) \in A_{\mathcal{T}}, \tag{3}$$

$$x_{ij}^{kt\bar{t}} \in \{0,1\} \quad \forall ((i,t),(j,\bar{t})) \in A_{\mathcal{T}} \cup H_{\mathcal{T}}, k \in K, \tag{4}$$

$$y_{ij}^{t\bar{t}} \in \mathbb{N}, \quad ((i,t),(j,\bar{t})) \in A_{\mathcal{T}}. \tag{5}$$

The objective function (1) is the sum of the fixed transportation costs as well as any variable costs associated with commodities transported by vehicles. Note that as the network spans both the line-haul and local delivery routes, the transportation costs include costs at both levels, as well. Constraints (3) ensure that each commodity is routed from origin to destination through the time–space network. Constraints (4) ensure sufficient capacity for each commodity move.

To formulate the SNDRP correctly, we must ensure that we accurately model the routing, and corresponding costs, of shipments at both the line-haul and local delivery routes. At the line-haul level, we must ensure that the shipment is routed from the supplier location to the destination terminal and that the appropriate costs are paid. This is modeled correctly (under the usual assumptions) by formulation (1)–(5). However, the local delivery routes present further additional features that we must model. Specifically, we must ensure that deliveries are made by vehicle routes and that the capacity of each vehicle is satisfied. As such, we next discuss how we construct a time–space network based upon known routes, such that our problem can be formulated directly as an SNDP. After that, we discuss how we adapt the variant of the SNDP presented above to our problem with an arc-based VRP-type formulation.

4.3 Route-based formulation

In the route-based formulation, denoted $\mathcal{F}_R$, we presume a known set of routes. We add a fourth set of (virtual) nodes, N_r, with each such node representing a local delivery route. Specifically, for each BB i, we assume a set of routes R_i that visit customers in $N_{\mathrm{c}}(i)$. For each such route, r, we add the node v_r to N_R. Our network then consists of the node set $N_{\mathcal{F}_{\mathcal{R}}} = N_{\mathrm{s}} \cup N_{\mathrm{bb}} \cup N_{\mathrm{c}} \cup N_R$.

We then let A_R denote the set of arcs connecting these virtual nodes to the rest of the network. Specifically, for each route $r \in R_i$ departing from breakbulk i, we add two types of arcs to the set A_R. The first is an arc of the form (i, v_r), from the breakbulk to the node representing the route. Assuming that the transportation cost of the route is given by c_r, we associate with arc (i, v_r) the fixed cost $f_{iv_r} = c_r$ and the variable cost $c_{iv_r} = 0$. We treat these arcs as uncapacitated and associate a travel time of 0 with each such arc. The second type of arc we add to A_R connects the node modeling the route to the customers that it serves. Specifically, given a route $r \in R_i$ that delivers commodities $k \in K_r$, we add the arc (v_r, d_k) to A_R. We associate with this arc $f_{v_r d_k} = c_{v_r d_k} = 0$. We associate with these arcs the travel time $tt_{v_r d_k}$ which is equal to the sum of travel and service times on route r between its departure from breakbulk i and its arrival on d_k. We then define the set $A_{\mathcal{F}_{\mathcal{R}}} = A \cup A_R$.

Note that given a route $r \in R_i$ that delivers commodities in the set $K_r \subseteq K$, by examining the corresponding deadlines l_k, we can compute a latest departure time, τ_r, for that route from the breakbulk i. In this formulation, when representing this route in the time–space network, we only consider nodes of the form (v, t) where $t \leq \tau_r$.

This modeling of the delivery routes is illustrated below for the problem of Fig. 2c.

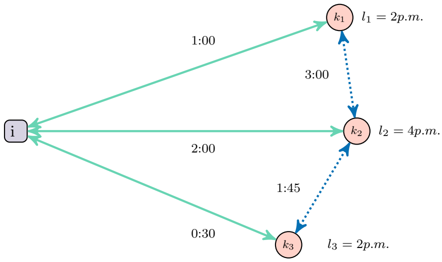
In this example, we first presume three commodities, k_1, k_2, and k_3, associated with breakbulk i. For the sake of readability, the respective destinations of these commodities are also denoted by k_1, k_2, and k_3. The times at which shipments to those destinations must be delivered are given in Fig. 2a. Similarly, Fig. 2b gives the travel time matrix between the breakbulk i and the destinations of all commodities.

Commodities	l_k
k_1	2 p.m.
k_2	4 p.m.
k_3	2 p.m.

(a) Commodities

tt	i	k_1	k_2	k_3
i	0	1:00	2:00	0:30
k_1	1:00	0	3:00	-
k_2	2:00	3:00	0	1:45
k_3	0:30	-	1:45	0

(b) Travel times between customer locations (in hours)



(c) Network

Fig. 2 Example of network creation: data of the problem

Next, we model three potential delivery routes. In Fig. 3a, we present the sequence of commodities delivered by each route and the corresponding latest departure time for that route from the depot. As an example of this calculation, route r_1 must deliver commodity k_2 by 4 p.m. but it takes 1 h to travel from i to the delivery location of k_1 and then 3 h between the delivery points of k_1 and k_2. Thus, the latest time at which the route can depart from i is 12 p.m. We illustrate the network that we create with virtual nodes in Fig. 3b. By constructing a network in this manner, we are able to model the SNDRP as a classical Service Network Design Problem defined on graph $D_{\mathcal{F}_{\mathcal{R}}} = (N_{\mathcal{F}_{\mathcal{R}}}, A_{\mathcal{F}_{\mathcal{R}}})$.

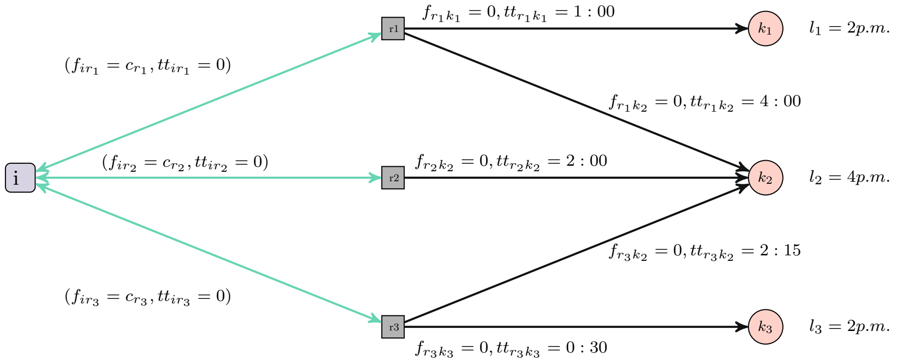
4.4 Arc-based formulation

We create a set of *local arcs*, denoted A_L. This set contains arcs of the form $(\mathrm{bb}(j), j)$, where $\mathrm{bb}(j) \in N_{\mathrm{bb}}$ and $j \in N_{\mathrm{c}}$, that model a move from a breakbulk to a customer location. It also includes arcs (i, j) between two customers, such that $\mathrm{bb}(i) = \mathrm{bb}(j)$. We then define $A_{\mathcal{F}_{\mathcal{A}}} = A \cup A_L$ and $N_{\mathcal{F}_{\mathcal{A}}} = N_{\mathrm{s}} \cup N_{\mathrm{bb}} \cup N_{\mathrm{c}}$. The arc-based formulation of the SNDRP, denoted $\mathcal{F}_{\mathcal{A}}$, is defined on graph $D_{\mathcal{F}_{\mathcal{A}}} = (N_{\mathcal{F}_{\mathcal{A}}}, A_{\mathcal{F}_{\mathcal{A}}})$, In addition, we let $A_{L\mathcal{T}}$ represent the arcs corresponding to A_L in the time–space network and $H_{L\mathcal{T}}$ represent the set of waiting arcs for each customer in N_{c}.

We then add constraints to the SNDP formulation (1)–(5) presented above to ensure that the vehicle moves defined by the value of variables $y_{ij}^{t\bar{t}}$, $(i,j) \in A_L$ form feasible routes. In our time–space network, a vehicle route begins at a breakbulk

r	Commodity	τ_r
r_1	k_1, k_2	12:00 p.m.
r_2	k_2	2:00 p.m.
r_3	k_3, k_2	1:30 p.m.

(a) Routes R_i


$f_{r_1k_1} = 0, tt_{r_1k_1} = 1:00$
r1
k_1
$l_1 = 2p.m.$
$(f_{ir_1} = c_{r_1}, tt_{ir_1} = 0)$
$f_{r_1k_2} = 0, tt_{r_1k_2} = 4:00$
i
$(f_{ir_2} = c_{r_2}, tt_{ir_2} = 0)$
r2
$f_{r_2k_2} = 0, tt_{r_2k_2} = 2:00$
k_2
$l_2 = 4p.m.$
$f_{r_3k_2} = 0, tt_{r_3k_2} = 2:15$
$(f_{ir_3} = c_{r_3}, tt_{ir_3} = 0)$
r3
k_3
$l_3 = 2p.m.$
$f_{r_3k_3} = 0, tt_{r_3k_3} = 0:30$

(b) Network with virtual nodes

Fig. 3 Enumerated routes and resulting network

$b \in N_{\text{bb}}$, and then take at path of arcs of the form $((i,t),(j,\bar{t}))$, where $i,j \in N_{\text{c}}(b)$ and $\bar{t} > t$.

We illustrate such a route in Fig. 4a, with details regarding the commodities to deliver in Fig. 4a. In this figure, the route begins at (bb_1, t_1), delivers commodities k_1, k_2, and k_3, and then returns to the associated breakbulk at time t_9.

To ensure that the values of the $y_{ij}^{t\bar{t}}$ variables form such a path, we add the following two constraints to the SNDP formulation (1)–(5):

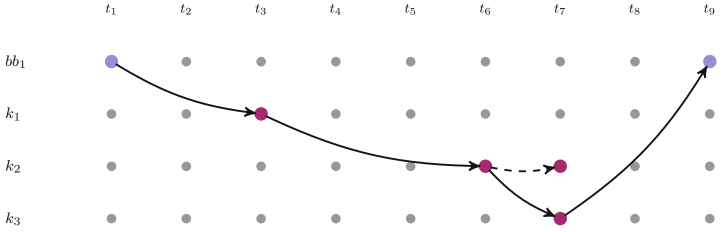
$$\sum_{((j,\bar{t}),(i,t))\in A_{\mathcal{LT}}\cup H_{\mathcal{LT}}} y_{ji}^{\bar{t}t} = \sum_{((i,t),(j,\bar{t}))\in A_{\mathcal{LT}}\cup H_{\mathcal{LT}}} y_{ij}^{t\bar{t}} \quad \forall i \in N_{\text{c}}, (i,t) \in N_{\mathcal{T}} \tag{6}$$

$$\sum_{t\in T_i} \sum_{((j,\bar{t}),(i,t))\in A_{\mathcal{LT}}} y_{ji}^{\bar{t}t} = \sum_{t\in T_i} \sum_{((i,t),(j,\bar{t}))\in A_{\mathcal{LT}}} y_{ij}^{t\bar{t}} \quad \forall i \in N_{\text{bb}}. \tag{7}$$

Constraints (6) ensure that every arrival at a customer location is followed by a departure to another node. These constraints ensure that the variables form a path for each vehicle. Constraints (7) enforce that the number of routes starting from a breakbulk equals the number of routes arrival at the same breakbulk. Recall that distribution routes have only two types of arcs in the physical network: (1) arcs between customers associated with the same breakbulk and (2) arcs between a customer and its associated breakbulk. An arc leaving a customer either goes to another customer associated with the same breakbulk or to the breakbulk itself. Hence, a path that starts at a breakbulk ends at the same breakbulk.

Note that constraints (6) and (7) do not forbid a path that consists solely of moves between customers (i.e., sub-tours on the physical network). Nevertheless, constraints (3) impose a path for each commodity between its origin and its destination. Hence, sub-tours never appear in an optimal solution.

Finally, we comment on how the formulation models the use of vehicles. Specifically, we note that constraints (4), which link the flow of commodities to that of vehicles, are only defined over arcs, $A_{\mathcal{T}}$ are only defined on arcs that model physical transportation. As a result, the model allows a commodity to flow on a "waiting" arc of $H_{\mathcal{LT}}$ without an accompanying vehicle. For example in Fig. 4a, commodity k_2 is delivered to the customer one period early (t_6). The optimization model allows for commodity k_2 to travel on a waiting arc to its destination (k_2, t_7) without the use of a vehicle, with the vehicle continuing on its route to (k_3, t_9).



(a) Vehicle route in time-space network that begins at bb_1

k	d_k	l_k
1	k_1	t_3
2	k_2	t_7
3	k_3	t_9

(b) Commodities

Fig. 4 Arc-based formulation

Finally, we note that the capacity of the vehicles on local delivery routes is modeled with the capacities on the arcs $(i,j) \in A_L$.

4.5 About model sizes

The SNDRP has the SNDP as a special case and is, therefore, NP-Hard. Both the route-based formulation and the arc-based formulation suppose that local deliveries are modeled on a time-expanded network. They necessitate a thinner level of discretization than is usually needed in SNDP at the tactical level. Hence, the theoretical models sizes make them too high to be solved by a solver. In this paper, both models are solved with the DDD algorithm which is introduced in the following section. DDD performs a progressive discretization of time which allow to reduce the size of the models that are actually sent to the solver.

5 Solution approach

Historically, when modeling with a time–space network, the choice of time points to model at each location was made a priori, and represented a trade-off between model size and accuracy. For example, to model activities during a day, one could choose to have two nodes, one that models activities from midnight to noon and another that models activities from noon until midnight. This approach has the advantage of only requiring two nodes, but may not accurately model when exactly activities occur. This is particularly important in a setting wherein consolidation is critical, as consolidation occurs not only in space but in time. If two shipments are transported from terminal A to terminal B, with the first having to depart before 9 am and the second after 11 am, then consolidation cannot occur. Yet, in our two-node discretization of a day, both would depart from the same node, and an SNDP defined on that time–space network would perceive them as consolidating. Another choice is to model the day with 24 nodes, one for each hour (e.g., (A, 0:00), (A, 1:00), …). Such an approach leads to a network and resulting SNDP that is significantly larger. However, the SNDP defined on that network will have a more accurate perception of consolidation opportunities.

The recent work of Boland et al. (2017) has shown that when solving SNDPs, one can instead determine this discretization of time dynamically during the search for an optimal solution in both space and time. The algorithm presented in that paper referred to as a Dynamic Discretization Discovery (DDD) scheme requires three inputs: (1) a network representing physical locations and transportation between them, (2) a set of shipments, each with an available and due time that must be routed through that network, and (3) a smallest possible time window to model at each node. As an example, a smallest window of time could be 5 min, meaning that we ideally wish to solve a SNDP defined on a time–space network that contains the nodes (A, 9:00), (A, 9:05), (A, 9:10), etc. for each location A. Such a problem is referred to as the Continuous Time Service Network Design Problem (CTSNDP). As such, the DDD algorithm will produce a solution to the CTSNDP time problem without enumerating all time points a priori. Specifically, its output is a route for each shipment both in space (the physical transportation moves) and time (when it dispatches from each terminal). As time is a critical feature of the SNDRP, we seek to leverage the capabilities of this algorithm (Fig. 5).

We do not describe in this paper the workings of the DDD algorithm in detail. The interested reader should consult (Boland et al. 2017). Instead, we will describe the steps the algorithm performs at a high level. Fundamentally, the algorithm is relaxation-based, as it repeatedly solves a relaxation of the CTSNDP, checks to see if the solution to the relaxation can be converted to a solution to the CTSNDP of the same cost, and if not refines the relaxation. If it can, the algorithm terminates, as it has found a solution that is optimal. A flow chart of the algorithm is given in Fig. 6.

That said, the relaxation is in itself an instance of the SNDP defined over what the authors refer to as a *partially time-expanded network*. A partially time-expanded network is one where locations are not all represented at every point in time; an example is given in Fig. 7, wherein location j is only represented at time point 1, location k at time points 2 and 3, and location l at time point 2. By starting with a relatively small partially time-expanded network, and carefully refining it at each iteration, the algorithm avoids the computational challenges associated with performing a full enumeration of time points at locations.

However, for the resulting SNDP to be a relaxation of the CTSNDP, care must be taken when constructing a partially time-expanded network. Boland et al. (2017) prove the theoretical properties, but fundamentally, the partially time-expanded network must be created in such a way that all paths that are feasible for the CTSNDP have a corresponding path in the partially time-expanded network whose travel time, as modeled by the nodes in the network, is shorter. Such paths are called *Early arrival paths*. To further explain, we illustrate in Fig. 8a a network with four nodes, and three arcs, with the travel times listed below each arc. Then, in Fig. 8b, we depict two early arrival paths. The first, from $i \rightarrow j \rightarrow k$ should take $2+2=4$ units of time, yet the network models it as taking only 3 as the arc ((j, 3), (k, 4)) underestimates the actual travel time of arcs of the form $((j,t),(k,t+2))$. Similarly, the path from $j \rightarrow k \rightarrow l$ is also an early arrival path as again, arc ((j, 3), (k, 4)) is "too short". Essentially, and in terms of the notation we have already introduced, to create a partially time-expanded network that induces an SNDP that is a relaxation of the CTSNDP, one must ensure that for every node $(i,t) \in N_{\mathcal{T}}$ and every arc $(i,j) \in A$, there must be an arc $((i,t),(j,t')) \in A_{\mathcal{T}}$, such that $t' - t \leq tt_{ij}$.

Next, we illustrate how a partially time-expanded network is refined when a solution to the resulting relaxation cannot be converted to a solution to the CTSNDP of equal cost. We first give in Fig. 9a the commodities that are to be routed, and in Fig. 9b, we illustrate the physical network over which they are to be routed. We presume that all arcs have capacity u_{ij} = 5, fixed cost f_{ij} = 10, and zero variable cost

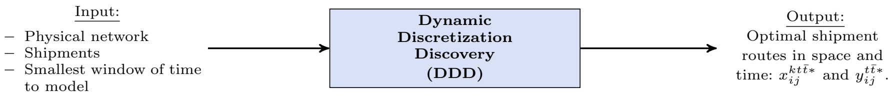

Input:
– Physical network
– Shipments
– Smallest window of time to model
Dynamic Discretization Discovery (DDD)
Output:
Optimal shipment routes in space and time: $x_{ij}^{kt\bar{t}*}$ and $y_{ij}^{t\bar{t}*}$.

Fig. 5 Use of DDD algorithm

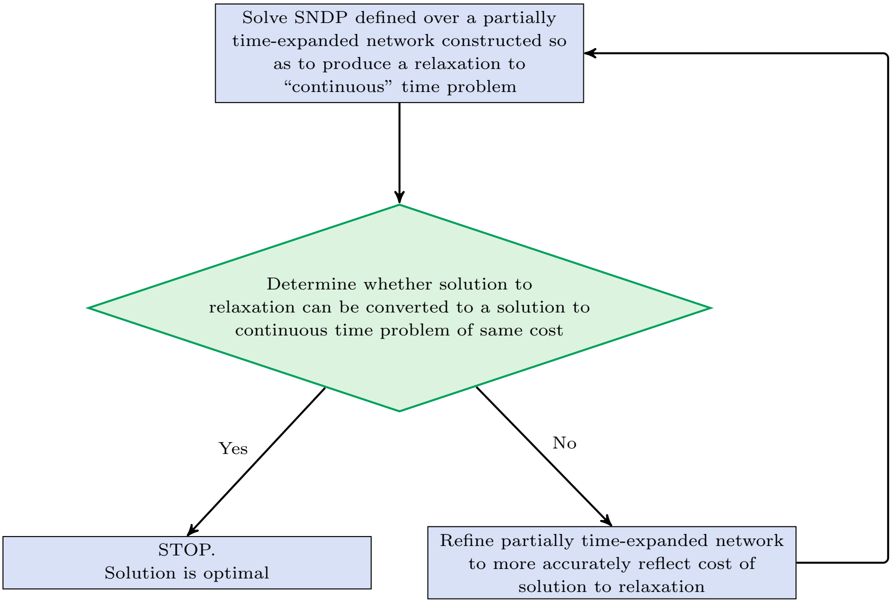
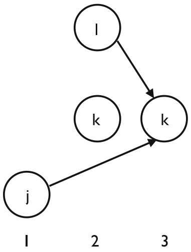
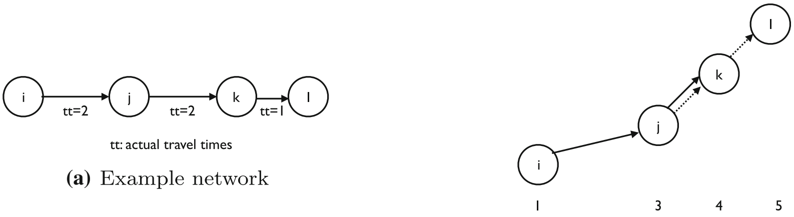

Solve SNDP defined over a partially time-expanded network constructed so as to produce a relaxation to “continuous” time problem
Determine whether solution to relaxation can be converted to a solution to continuous time problem of same cost
Yes
No
STOP. Solution is optimal
Refine partially time-expanded network to more accurately reflect cost of solution to relaxation

Fig. 6 DDD flow chart

Fig. 7 Partially time-expanded network





(a) Example network

(b) Early Arrival paths

Fig. 8 Example network and early arrival paths

($c_{ij} = 0$). We then illustrate in Fig. 9c a solution to the relaxation SNDP at an iteration.

We observe that in this solution to the relaxation, commodities k_1 and k_2 are able to consolidate on the arc ((j, 3), (l, 6)), meaning that the fixed charge for arc (j, l)

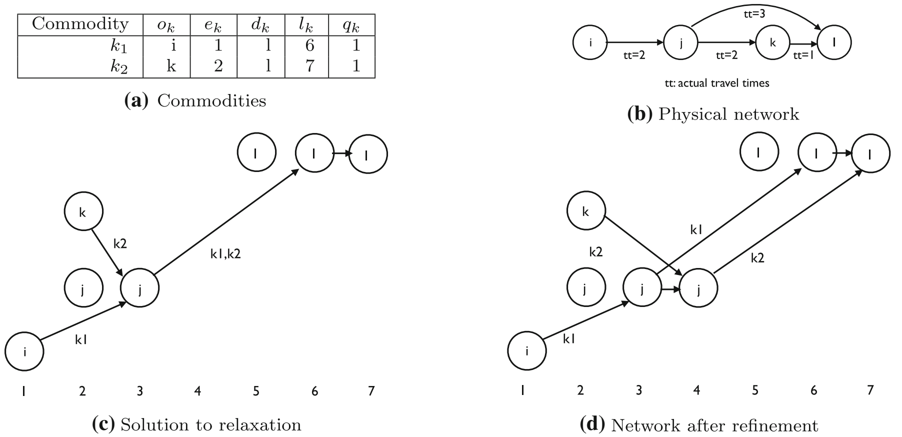
need only be paid once. We next note that because k_1 must arrive at l by time period 6, and arc (j, l) has travel time of 3, k_1 cannot depart node j later than time period 3. At the same time, we note that as commodity k_2 becomes available at time 2 and the arc (k, j) has travel time 2, k_2 cannot arrive at node j until time period 4. Thus, in this partially time-expanded network, arc ((k, 2), (j, 3)) being too short is enabling a consolidation on arc ((j, 3), (l, 6)) that is not truly achievable. Thus, this solution cannot be converted to one to the CTSNDP of the same cost.

The reason why arc ((k, 2), (j, 3)) is too short is because there is no node in the current partially time-expanded network that enables us to model the actual travel time. Thus, this is how the algorithm refines the network. It discovers an arc that is enabling a *phantom* consolidation (e.g., the consolidation on arc ((j, 3), (l, 6)) in our example), because it is too short and then adds a new node to the partially time-expanded network that enables the true travel time of the arc to be modeled. In Fig. 9d, we see that the node (j, 4) has been added, which enables the arc ((k, 2), (j, 3)) to be replaced with the arc ((k, 2), (j, 4)) which is the correct length. Similarly, as arc $(j, l) \in A$, arc ((j, 4), (l, 7)) is added to the network. Now, the consolidation that occurred in the solution to the previous relaxation cannot occur.

We refer the interested reader to Boland et al. (2017) to see details, regarding how to discover such phantom consolidations as well as the arcs that enable them, and methods for converting solutions to relaxations to solutions of CTSNDP of potentially greater cost, but still high-quality. However, fundamentally, we are able to use this algorithm as a black box for solving our problem, and we next discuss our computational experiences doing so.

6 Computational results

This section presents the computational results for both formulations $\mathcal{F}_R$ and $\mathcal{F}_A$. We first describe the instances generated for these experiments. Then, we compare



Commodity	o_k	e_k	d_k	l_k	q_k
k_1	i	1	1	6	1
k_2	k	2	1	7	1

(a) Commodities

(b) Physical network

(c) Solution to relaxation

(d) Network after refinement

Fig. 9 Example of refining a partially time-expanded network

the performance of the DDD algorithm on each formulation, and finally, we validate the integration of the SNDP and VRP.

6.1 Experimental setting

Both the geographic and temporal aspects of the instances used in our computational study are based on the logistics network of a French 3PL named 4S Network. In this setting, customer locations correspond to distribution platforms for retail stores and supplier locations correspond to areas wherein there are a large number of factories. The 4S Network breakbulk facilities are called Collaborative Routing Centers (CRCs).[1]

To be precise, we generate instances with the following characteristics:

- The total number of commodities is either 25, 50, 100, or 200. We recall that a customer can receive only one commodity; hence, there are as many customer locations as commodities. Ten sets of commodities are created for each instance size, which gives $10 \times 4 = 40$ different commodity sets.
- The width of the time windows $[e_k, l_k]$ either varies between 12 and 24 h (tight time windows category) or between 24 and 36 h (wide time windows category), according to the distance between the origin and destination of the commodities.
- For each commodity set, four instances are generated, using different number of BB (4, 6, 8, and 10). To allow a better comparison, the logistics network which integrates 4 BB is included in the logistics network with 6 BB, etc. This permits us to observe the costs variations related to the number of BB in the network, although customers assignment may differ when the number of BB increases.

Thus, $40 \times 2 \times 4 = 320$ instances are generated. Instances are available at http://www.vrp-rep.org/datasets/item/2017-0030.html and the solutions are available upon request.

To limit the number of variables in the formulation $\mathcal{F}_R$, the set of vehicle routes has been restricted. As generally observed in the practical application that motivated this study, the maximal number of customers per route is limited to four. Hence, in our experimentations, both formulations are not strictly equivalent and the resolution of $\mathcal{F}_R$ gives an upper bound of the resolution of the SNDRP.

Regarding our experimental setting, we execute the DDD algorithm on each instance of each formulation for a maximum of 2 h and an optimality gap of 1%. We also allowed each execution of the algorithm at most 1 GB of RAM. We note that DDD's guarantee of convergence is predicated upon the ability to build and solve an MIP at each iteration. As a result, it is possible that with an MIP exceeding the memory limit allowed, the algorithm can terminate without finding a feasible solution. In the next section, we first study the ability of the DDD algorithm to solve instances of each formulation. Finally, recall that the problem we seek to solve integrates two problems, the SNDP and the VRP, that are traditionally solved in a separate and sequential manner (i.e., SNDP then VRP). Thus, we finish the next section by validating the use of the integrated problem we propose to solve. To do

[1] http://www.crc-services.com/.

so, we compare the quality of the solution produced when solving the arc-based formulation of this problem with the one produced by solving the separate problems in a sequential manner.

6.2 Ability of DDD to solve route and arc-based formulations

We first study the ability of the DDD algorithm to solve instances of the route-based formulation, $\mathcal{F}_R$. Specifically, we report in Fig. 1, for each value of the number of commodities and of the number of CRC, the following four statistics: (1) the percentage of instances for which the algorithm was able to produce a feasible solution (i.e., did not exceed the memory limit when solving an MIP at an iteration), (2) the percentage of instances it could solve to optimality, (3) the average amount of time, in seconds, that DDD required to solve those instances to optimality, and (4) the optimality gap DDD reported at termination, averaged over all instances.

We see that while $\mathcal{F}_R$ allows us to model rich rules governing local delivery routes, it is very difficult to solve. This is primarily due to the large size of these instances that results from enumerating routes a priori. We see that the number of commodities has a significant impact on the ability of DDD to solve an instance. That said, we see that the number of CRCs also has an impact on DDDs

Table 1 Computational results with the route-based formulation ($\mathcal{F}_R$)

# Commodities	# CRCs	% Feasible	% Optimal	Time to optimal (s)	Optimality gap (%)
25	4	100	100	240	0.86
	6	100	100	205	0.88
	8	100	83	2247	1.54
	10	100	5	5106	7.22
	Average	100	72	1895.5	2.63
50	4	78	78	1084	0.94
	6	80	23	2983	3.44
	8	85	0	–	9.42
	10	85	0	–	25.28
	Average	82	25.25	2033.5	9.77
100	4	50	18	1971	1.77
	6	50	0	–	7.30
	8	50	0	–	14.58
	10	50	0	–	24.91
	Average	50	4.5	1971	12.14
200	4	25	3.00	774	2.32
	6	25	0	–	6.87
	8	25	0	–	14.76
	10	25	0	–	20.23
	Average	25	0.75	774	11.05

performance, as the percentage of instances with 25 commodities that it can solve decreases as the number of CRCs increases. Similarly, we see that for instances with four CRCs, DDD is able to produce a solution of relatively high quality in a reasonable run-time.

We next turn our attention to the ability of DDD to solve instances of the arc-based formulation, $\mathcal{F}_A$. Table 2 presents the same statistics reported in Table 1, however, derived from solving instances of formulation $\mathcal{F}_A$.

We see that, on instances of this formulation, DDD produces a feasible solution for every instance, and solves many of them in a short amount of time. We also note that overall but the largest instances DDD is able to produce a solution with a small optimality gap. Like the route-based formulation, DDD's performance depends on both the number of commodities and the number of CRCs. While the percentage of instances for a given number of commodities that are solved to optimality remains roughly the same as the number of CRCs increases, the time DDD needs increases.

We next turn our attention to the sensitivity of DDD's performance to commodity time window width when solving the arc-based formulation. Specifically, we report in Table 3 the % optimal and time to optimal statistics, averaged over instances with the same number of commodities and time window width.

Table 2 Computational results with the arc-based formulation ($\mathcal{F}_A$)

# Commodities	# CRCs	% Feasible	% Optimal	Time to optimal (s)	Optimality gap (%)
25	4	100	100	7	0.44
	6	100	100	8	0.42
	8	100	100	13	0.49
	10	100	100	20	0.37
	Average	100	100	12	0.43
50	4	100	100	75	0.51
	6	100	90	680	2.55
	8	100	80	1119	1.07
	10	100	60	808	1.75
	Average	100	82.50	670.5	1.47
100	4	100	80	1398	0.59
	6	100	40	56	3.15
	8	100	45	89	4.64
	10	100	45	145	7.54
	Average	100	52.5	422	3.98
200	4	100	50	901	1.55
	6	100	40	402	3.60
	8	100	45	670	8.49
	10	100	45	1223	13.59
	Average	100	45	799	6.81

We see that the performance of DDD is highly sensitive to the width of commodity time windows, as it is able to solve nearly all instances with 24 h time windows, regardless of the number of commodities in the instance. To be precise, DDD is able to solve 97.06% of instances that have a 24 h time window, and needs only 216.99 s, on average, to do so. However, DDD is only able to solve 48.57% of instances that have a 36 h time window, and needs 3358.76 s, on average, to do so.

We finish with a direct comparison of DDDs ability to solve instances of each formulation. We repeat in Table 4 the % optimal statistics from the previous tables, by number of commodities and number of CRCs, for each formulation. The results in Table 4 clearly show that DDD is able to solve many more instances of the arc-based formulation than the route-based formulation. It also highlights that the arc-based formulation is more robust with respect to the number of CRCs. While DDDs performance when solving instances of the arc-based formulation does degrade as the number of CRCs increases, it does so at a much slower rate than when solving instances of the route-based formulation.

As mentioned earlier, one drawback of the route-based formulation is that it relies on an enumeration of routes, which in turn greatly increases instance size. To study that further, we report, in Table 5, the size, in terms of numbers of variables and constraints, of the last integer program solved by DDD in the course of solving an instance of each formulation. We see that the route-based formulation does, indeed, induce much larger integer programs, which in turn hampers DDDs ability to solve instances of that formulation. We also see that the differences in numbers of variables and constraints grow in the number of commodities.

6.3 Validating the SNDRP

We finish by validating the solution of the SNDRP, which integrates the decisions prescribed independently by the SNDP and the VRP. To do so, we consider the arc-based formulation of the SNDRP and decompose an instance of this formulation into one SNDP instance and one VRP instance. We solve the SNDP instance (with DDD) to determine the path each commodity follows through the line-haul network. These paths partially determine arrival times for commodities at their respective destination terminals, which can then be used to formulate and solve an instance of the VRP.

Table 3 Sensitivity of DDD performance to time window width

Time window width	25 commodities		50 commodities		100 commodities		200 commodities	
	% Optimal	Time to optimal (s)	% Optimal	Time to optimal (s)	% Optimal	Time to optimal (s)	% Optimal	Time to optimal (s)
12–24 h	100	2.25	89	194.13	100	80.89	100	624.66
24–36 h	100	21.70	56	3181.48	33	6369.80	0	3862.05
Average	100	11.98	72	1687.80	65	3434.97	50	2351.27

Table 4 % of instances DDD can solve, by formulation

# Commodities	4 CRCs		6 CRCs		8 CRCs		10 CRCs	
	$\mathcal{F}_R$ (%)	$\mathcal{F}_A$ (%)	$\mathcal{F}_R$ (%)	$\mathcal{F}_A$ (%)	$\mathcal{F}_R$ (%)	$\mathcal{F}_A$ (%)	$\mathcal{F}_R$ (%)	$\mathcal{F}_A$ (%)
25	100	100	100	100	83	100	5	100
50	78	100	23	90	0	80	0	60
100	18	80	0	40	0	45	0	45
200	3	50	0	40	0	45	0	45
Average	49.75	82.50	30.75	67.50	20.75	67.50	1.25	62.50

Table 5 Sizes of final integer programs solved by DDD, by formulation

# Commodities	# Variables		# Constraints	
	$\mathcal{F}_R$	$\mathcal{F}_A$	$\mathcal{F}_R$	$\mathcal{F}_A$
25	6631.56	32,717.24	3153.25	9425.99
50	33,874.11	118,824.44	11,733.85	34,516.38
100	103,315.24	187,360.81	40,470.01	63,003.74
200	234,868.75	336,042.75	121,096.76	175,824.45

That said, these commodity arrival times constrain the feasible region of the resulting VRP, as they, along with the commodity due times, dictate the amount of time available for routing the commodity to its destination. As the SNDP does not recognize the impact of these arrival times on the cost of the resulting VRP, we solve an auxiliary optimization problem that determines dispatch times for each commodity on each arc in its respective path in such a way that it arrives at its destination as early as possible, while the consolidations in the optimal solution to the SNDP are maintained. As such, we create an alternative optimal solution to the SNDP that allows for greater flexibility in the resulting VRP instance. This instance of the VRP is also solved with DDD (i.e., it is formulated as an arc-based formulation of the SNDP). We present the auxiliary optimization problem that we solve in Appendix A.

The solutions to the SNDP and VRP instances are then combined to construct a feasible solution to the SNDRP, with objective function value $z_{\text{SNDP+VRP}}$. We compare this objective function value with the objective function value of a solution to the SNDRP produced by DDD, denoted z_{SNDRP}. Specifically, we compute the gap $(z_{\text{SNDP+VRP}} - z_{\text{SNDRP}})/z_{\text{SNDP+VRP}}$. We consider arc formulation instances with 25 or 50 commodities as we seek to validate the use of the SNDRP with optimal solutions to the SNDRP and those are the instances that DDD is most likely to solve. We report in Table 6a averages of these gaps, averaged over instances with the same number of commodities and time window width. We also report in Table 6b the average time DDD required to solve these instances. Specifically, the time DDD required to solve instances of the SNDRP as well as the sum of the time DDD required to solve an instance of the SNDP and the time required to solve the corresponding VRP instance.

We see that solving the integrated problem yields nearly 3% in savings, on average. Not surprisingly, the savings are greater with instances that have narrower time windows, as those are the instances for which the paths through the line-haul network have the greatest impact on the feasible region of the local delivery routes. We also see that the savings increase, on average, as the number of commodities increases, suggesting that similar if not greater savings may exist for larger instances. We also see that the SNDRP is a harder problem to solve, as the time required to do so is greater. The results also suggest that this increase in computation time may also increase with the number of commodities. From these results, we conclude that the SNDRP is worth solving, particularly given that it may be solved in settings wherein planning is done for a weekly (if not shorter) time period, and thus a 3% savings can yield significant annual savings. However, heuristic methods may be necessary for such situations where run-times appropriate for practice are short.

7 Conclusions and future work

In this paper, we introduced a new optimization problem that integrates a Service Network Design Problem (SNDP) on a line-haul network, and one Vehicle Routing Problem (VRP) in each local distribution networks starting from a breakbulk location. We called this problem the Service Network Design and Routing Problem (SNDRP).

We proposed two solution approaches for this problem. Both involve formulating the problem as a variant of the SNDP, and then solving that variant with the

Table 6 Comparison of solving SNDRP and SNDP + VRP

(a) Savings produced by solving SNDRP vs SNDP + VRP

# Commodities	Time window width		
	24 h (%)	36 h (%)	Average (%)
25	1.70	1.31	1.51
50	4.33	3.78	4.05
Average	3.02	2.55	2.78

(b) Solve time for SNDRP and SNDP + VRP

# Commodities	SNDRP (s)	SNDRP + VRP (s)
25	20.81	13.65
50	1939.18	1117.79
Average	979.99	565.72

Dynamic Discretization Discovery (DDD) algorithm introduced in Boland et al. (2017). While similar in structure, the approaches differ in their modeling capabilities and computational complexity. The first, a route-based formulation, is based on a given set of potential local distribution routes, and thus enables the modeling of many real-life constraints and cost structures. However, similar to route-based formulations of the VRP, instances of this formulation suffer from an explosion in size when there are many potential routes, and thus are hard to solve for all, but the smallest instance sizes. That said, for a carrier that already has a reasonably small set of potential candidate routes, solving the route-based formulation may not be overly time-consuming. The second, an arc-based formulation, employs variables and constraints that are analogous to those seen in a compact VRP formulation. While this formulation is less amenable to modeling real-life considerations, it does not suffer from the same explosion in instance size, and thus is much easier to solve. For a carrier with operational constraints that cannot be modeled with this formulation, this arc-based solution approach could be used as a tool for determining shipment paths through the line-haul network as well as dispatch times for local delivery routes. These dispatch times as well as the shipments that are to be picked up for local delivery at those times could then be used as input to a more classical VRP-type model. Such an approach could yield savings for the carrier as it would still approximate the time needed for local delivery when determining shipment paths through the line-haul network, and thus produce shipment paths that are amenable to cost-effective delivery routes.

Several research directions emerge from this work. Regarding the route-based formulation, dynamically generating routes instead with column generation techniques could reduce the number of variables and thus enable the solution of larger instances. A second direction is to extend the modeling capability of the arc-based formulation. The prior assignment of each customer location to a unique breakbulk could be avoided by integrating new constraints stating that vehicle must start and finish their route at a same (yet to be defined) breakbulk. Another direction of great practical interest concerns modeling pickup and delivery routes instead of routes that only perform deliveries. Finally, a last direction concerns an integrated model for the location of breakbulks that recognizes the impact of those locations on routing through the line-haul network as well as local delivery areas.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant no. CMMI-1435625.

A Optimization problem solved to determine commodity arrival times in SNDP + VRP

In this section, we present the optimization problem that is solved in the SNDP + VRP solution approach described in Sect. 6.3. Recall that in that solution approach, an instance of the SNDRP is decomposed into an instance of the SNDP, which is solved to (partially) determine the corresponding instance of the VRP that must be solved. Specifically, the SNDP instance is solved to determine the paths for commodities through the line-haul network from their origin terminal to their destination terminal. While the corresponding instance of the VRP that must be solved is partially determined by the times at which commodities arrive at destination terminals, a given collection of commodity paths may yield sets of potential arrival times. As mentioned in Sect. 6.3, to implement SNDRP + VRP, we solve an auxiliary optimization problem which seeks to determine dispatch times for each commodity on each arc in its respective path in such a way that the commodity arrives at its destination as early as possible, while the consolidations in the optimal solution to the SNDP are maintained. We next present this optimization problem.

Having solved the SNDP instance for the optimal solution $(\bar{y}, \bar{x})$, we denote the path taken by commodity k in that solution by $p^k = \{i_0, \ldots, i_{m_k}\}$, where i_0 is $k's$ origin terminal and i_{m_k} its destination terminal. Namely, p^k consists of the arcs $(i,j) \in A$ and times $t \in T_i$, such that $\bar{x}_{ij}^{kt\bar{t}} = 1$. Next, for arc $(i,j) \in A, t \in T_i$, we let K_{ij}^t represent the set of commodities that dispatch at time t on arc (i, j) in the optimal solution to the SNDP. Formally, K_{ij}^t consists of all commodities k, such that $\bar{x}_{ij}^{kt\bar{t}} = 1$. We let the set T_{ij} denote the set of time points t at which arc (i, j) dispatches in the optimal solution to the SNDP, or, the set of time points t, such that $\bar{x}_{ij}^{kt\bar{t}} = 1$ for some $k \in K$. The set K_{ij}^t represents the consolidations that we seek to maintain. We let the continuous variable δ_{i_{v-1},i_v}^k denote the time at which commodity k dispatches on arc (i_{v-1}, i_v) in its path, p^k. Then, to determine the times at which commodities arrive at destination terminals, we solve the following optimization problem:

$$\text{minimize} \sum_{k \in K} \delta_{m_k-1,m_k}^k + tt_{m_k-1,m_k}^k$$

subject to

$$\delta_{i_0,i_1}^k \geq e_k \quad \forall k \in K, \tag{8}$$

$$\delta_{i_{v-1},i_v}^k + tt_{i_{v-1},i_v} \leq \delta_{i_v,i_{v+1}}^k \quad \forall v = 0, \ldots, m_k - 1, k \in K \tag{9}$$

$$\delta_{i_{m_k-1},i_{m_k}}^k + tt_{i_{m_k-1},i_{m_k}} \leq l_k \quad \forall k \in K, \tag{10}$$

$$\delta_{i,j}^k = \delta_{i,j}^{k'} \quad \forall (i,j) \in A, t \in T_{ij}, k, k' \in K_{ij}^t, \tag{11}$$

$$\delta^k_{i_{v-1},i_v} \geq 0 \quad \forall v = 0, \ldots, m_k - 1, k \in K. \tag{12}$$

The objective seeks to minimize the sum of times at which commodities arrive at destination terminals. Constraints (8) ensure that each commodity departs from its origin terminal after the time at which it is available. Constraints (9) ensure that the dispatch times observe arc travel times, whereas constraints (10) ensure that each commodity arrives at its destination before it is due. Constraints (11) ensure that if commodities travel on an arc at the same time in the optimal solution to the SNDP, they still do so with these new dispatch times. Finally, constraints (12) define the decision variables for the optimization problem and their domain. After solving this optimization problem for the dispatch times $\bar{\delta}^k_{i_{v-1},i_v}$, the times $\bar{\delta}^k_{m_k-1,m_k} + tt^k_{m_k-1,m_k}$ are the times at which commodities are available for delivery in the local distribution network.

References

Anderluh A, Hemmelmayr VC, Nolz PC (2017) Synchronizing vans and cargo bikes in a city distribution network. Cent Eur J Oper Res 25(2):345–376

Andersen J, Crainic TG, Christiansen M (2009a) Service network design with asset management: formulations and comparative analyses. Transp Res Part C Emerg Technol 17(2):197–207

Andersen J, Crainic TG, Christiansen M (2009b) Service network design with management and coordination of multiple fleets. Eur J Oper Res 193(2):377–389

Berbeglia G, Cordeau JF, Gribkovskaia I, Laporte G (2007) Static pickup and delivery problems: a classification scheme and survey. Top 15(1):1–31

Boland N, Hewitt M, Marshall L, Savelsbergh M (2017) The continuous-time service network design problem. Oper Res. https://doi.org/10.1287/opre.2017.1624

Cortés CE, Matamala M, Contardo C (2010) The pickup and delivery problem with transfers: formulation and a branch-and-cut solution method. Eur J Oper Res 200(3):711–724

Crainic TG (2000) Service network design in freight transportation. Eur J Oper Res 122(2):272–288

Crainic TG, Ricciardi N, Storchi G (2009) Models for evaluating and planning city logistics systems. Transp Sci 43(4):432–454

Crainic TG, Hewitt M, Toulouse M, Vu D (2016) Service network design with resource constraints. Transp Sci 50(4):1380–1393

Dellaert N, Saridarq FD, Van Woensel T, Crainic TG (2016) Branch and price based algorithms for the two-echelon vehicle routing problem with time windows. Les cahiers du CIRRELT (CIRRELT-2016-45)

Desaulniers G, Desrosiers J, Solomon MM (2006) Column generation. Springer Science and Business Media, Berlin

Dondo RG, Mendez CA (2014) A branch-and-price approach to manage cargo consolidation and distribution in supply chains. Ind Eng Chem Res 53(44):17226–17239

Drexl M (2012) Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints. Transp Sci 46(3):297–316

Erera A, Hewitt M, Savelsbergh M, Zhang Y (2013a) Improved load plan design through integer programming based local search. Transp Sci 47(3):412–427

Erera AL, Hewitt M, Savelsbergh MW, Zhang Y (2013b) Creating schedules and computing operating costs for LTL load plans. Comput Oper Res 40(3):691–702

Grangier P, Gendreau M, Lehuédé F, Rousseau LM (2016a) An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. Eur J Oper Res 254(1):80–91

Grangier P, Gendreau M, Lehuédé F, Rousseau LM (2016b) A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. Les cahiers du CIRRELT CIRRELT-2016-09

Guastaroba G, Speranza MG, Vigo D (2016) Intermediate facilities in freight transportation planning: a survey. Transp Sci 50(3):763–789

Jarrah AI, Johnson E, Neubert LC (2009) Large-scale, less-than-truckload service network design. Oper Res 57(3):609–625

Masson R, Lehuédé F, Péton O (2013a) An adaptive large neighborhood search for the pickup and delivery problem with transfers. Transp Sci 47:356–379

Masson R, Lehuédé F, Péton O (2013b) Efficient feasibility testing for request insertion in the pickup and delivery problem with transfers. Oper Res Lett 41(3):211–215

Mitrović-Minić S, Laporte G (2006) The pickup and delivery problem with time windows and transshipment. INFOR 44(3):217–228

Morais VW, Mateus GR, Noronha TF (2014) Iterated local search heuristics for the vehicle routing problem with cross-docking. Expert Syst Appl 41(16):7495–7506

Parragh SN, Doerner KF, Hartl RF (2008) A survey on pickup and delivery problems. Journal für Betriebswirtschaft 58(1):21–51

Pedersen MB, Crainic TG, Madsen OB (2009) Models and tabu search metaheuristics for service network design with asset-balance requirements. Transp Sci 43(2):158–177

Petersen HL, Ropke S (2011) The pickup and delivery problem with cross-docking opportunity. In: Proceedings of the second international conference on computational logistics, Springer, Berlin, Heidelberg, ICCL'11, pp 101–113

Qu Y, Bard JF (2012) A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment. Comput Oper Res 39(10):2439–2456

Santos FA, Mateus GR, da Cunha AS (2011) A novel column generation algorithm for the vehicle routing problem with cross-docking. In: Pahl J, Reiners T, Voss S (eds) Network optimization, vol 6701. Lecture notes in computer science. Springer, Berlin, pp 412–425

Santos FA, Mateus GR, da Cunha AS (2013) The pickup and delivery problem with cross-docking. Comput Oper Res 40(4):1085–1093

Tarantilis CD (2013) Adaptive multi-restart tabu search algorithm for the vehicle routing problem with cross-docking. Optim Lett 7(7):1583–1596

Toth P, Vigo D (2014) Vehicle routing: problems, methods, and applications, vol 18. SIAM, Philadelphia

Wen M, Larsen J, Clausen J, Cordeau JF, Laporte G (2008) Vehicle routing with cross-docking. J Oper Res Soc 60(12):1708–1718